Assembly free analysis with k-mers!!!!!

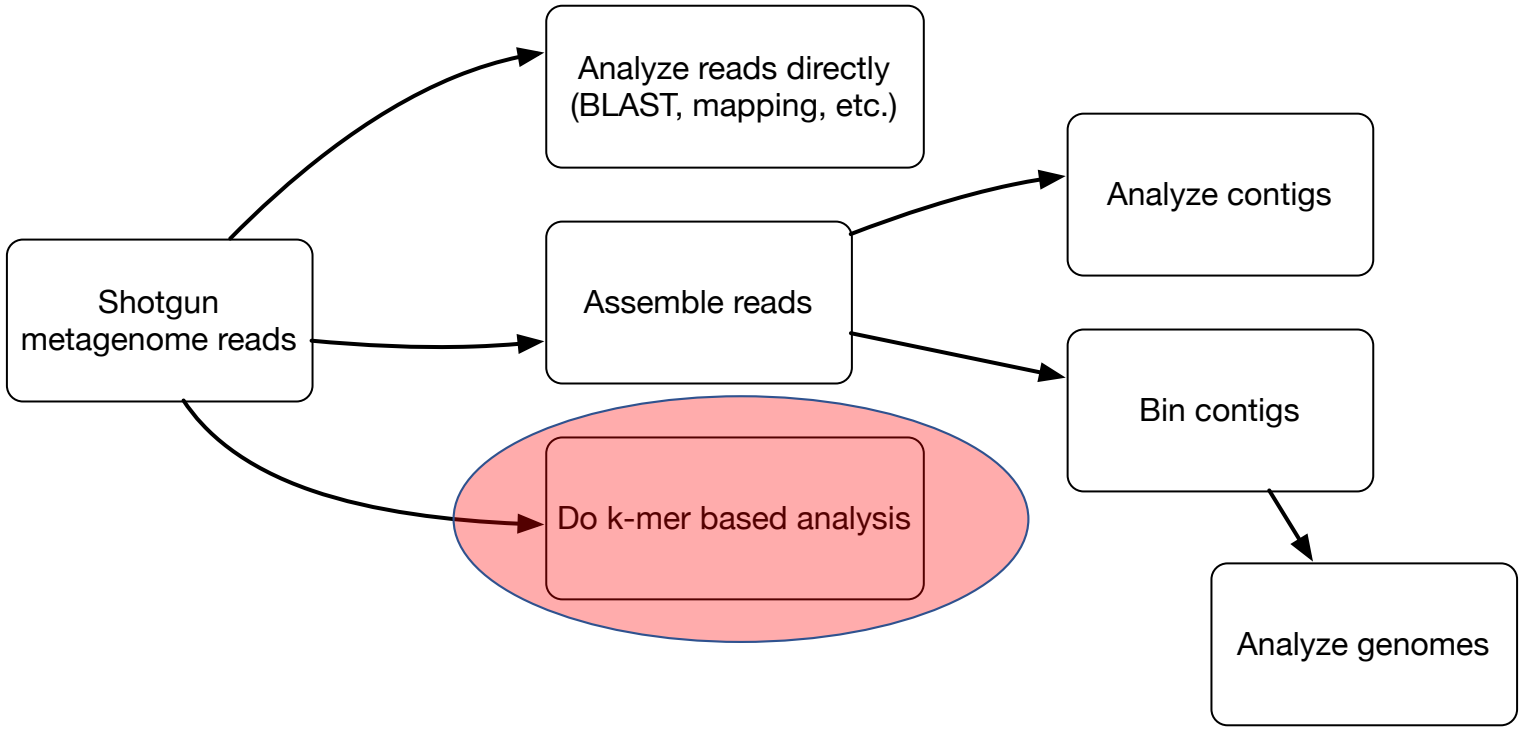
Titus Brown

MBL STAMPS 2022

July 26th

(Expurgated copy for during class; will be replaced with more complete copy, promise.)

Options for analyzing and *summarizing* shotgun metagenome content.



A "k-mer" is a word of DNA that is k long:

```
ATTG – a 4–mer
ATGGAC – a 6–mer
```

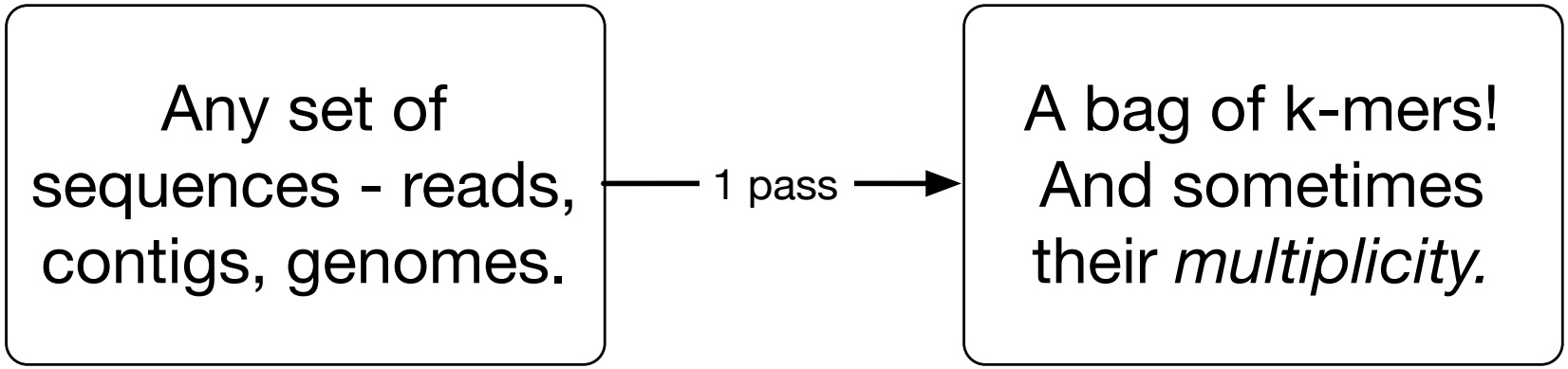
Typically we extract k-mers from genomic assemblies or read data sets by running a k-length window across all of the reads and sequences -- e.g. given a sequence of length 16, you could extract 11 k-mers of length six from it like so:

```
AGGATGAGACAGATAG
```

becomes the following set of 6-mers:

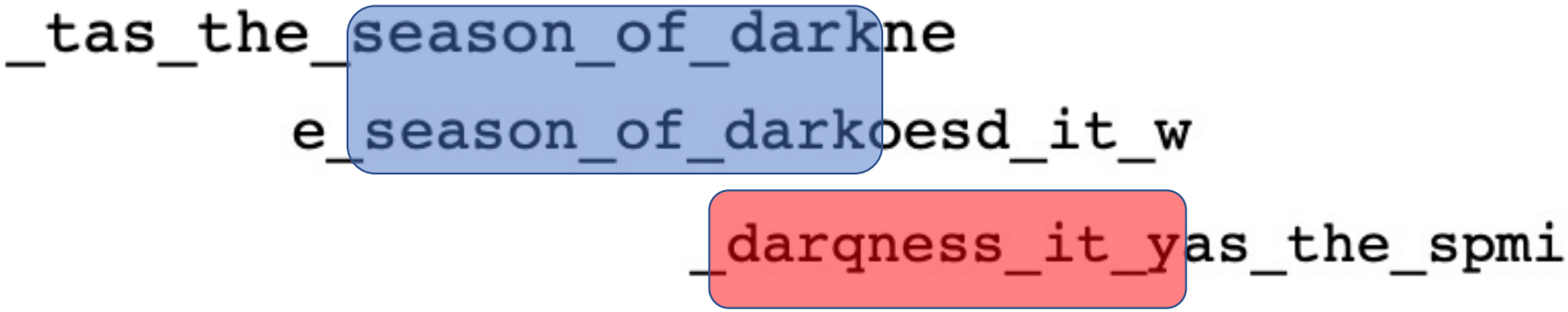
```
AGGATG
 GGATGA
  GATGAG
   ATGAGA
    TGAGAC
     GAGACA
      AGACAG
       GACAGA
        ACAGAT
         CAGATA
          AGATAG
```

K-mers can be extracted from *any* sequence!



e.g. “K-mer ATCCGATGACCAGATAGAGA is present 18 times in this metagenome”

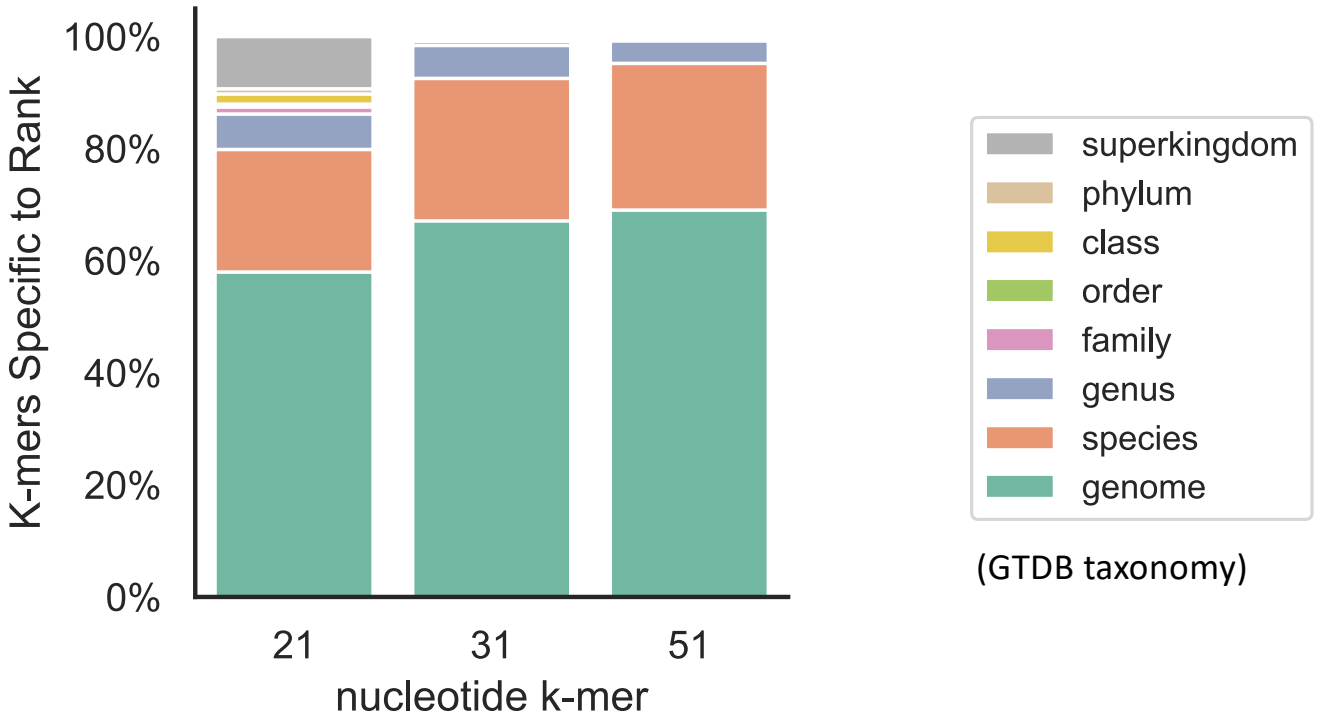
K-mers “stack” across reads – if no mismatches.



Errors typically represent as *unique* k-mers;
“Real” sequence typically has higher multiplicity

Prompt: how do you pick the size(s) of k to use to compare genomes and metagenomes?

31-mers are genome and species specific.



(GTDB taxonomy)

Pierce-Ward et al., in preparation.

Summary: nucleotide k-mers are very specific and very sensitive!

- ~60% of the time or more, a 31-mer will be unique to a bacterial or archaeal *genome.* 99.9% of the time it will be unique within genus!
  - Conversely, nucleotide k-mers are not useful for taxonomy analysis outside of genus 🤷‍♂️
- Sequencing error rarely yields a known k-mer, so if a known k-mer is present in a metagenome, it is almost certainly ”real” signal.
  - Can also require that a k-mer be present in multiple reads, which increases likelihood of it being real.

Next: K-mers are more powerful when they travel together!

What can you do with ***bags*** of k-mers?

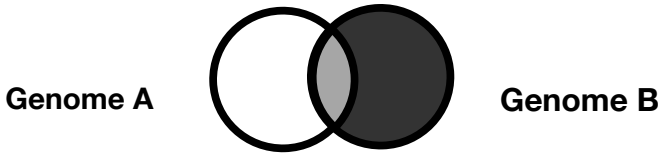
You can find and cluster genomes by similarity!

You can find reference genomes that match to a metagenome!

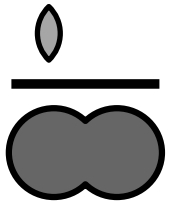
These two things alone turn out to be enough to be getting on with ;).

K-mer comparisons for Sequence Similarity: Jaccard

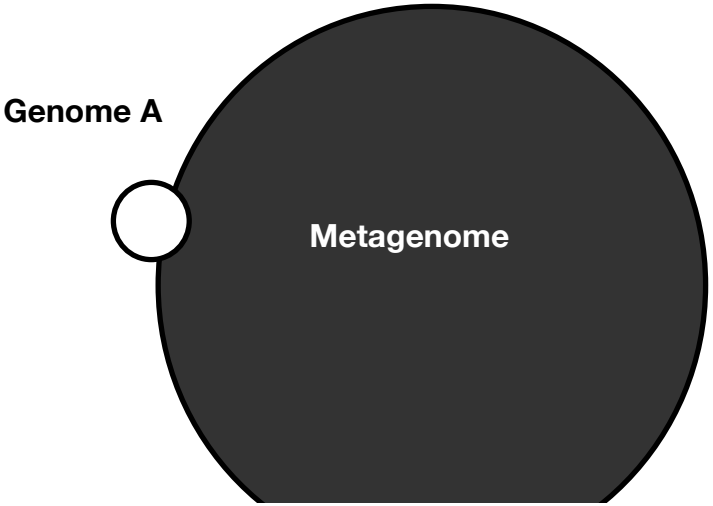
How similar are my genomes?



$$\text{Jaccard} = \frac{\text{intersection}}{\text{union}} =$$



Is genome A present in my Metagenome?



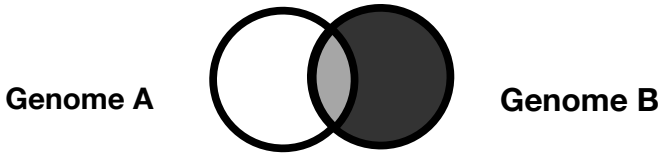
~~**Jaccard**~~

Metagenome likely contains several genomes: this comparison does not make sense

X

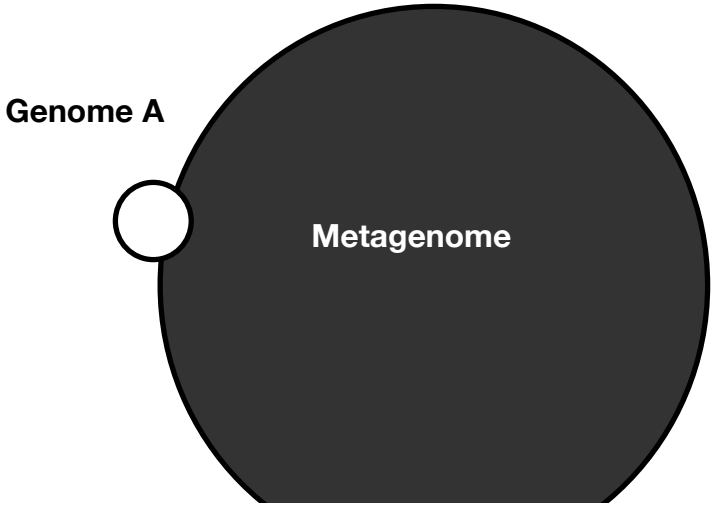
K-mer comparisons for Sequence Similarity: Containment

How similar are my genomes?



$C_{B \text{ in } A}$ $C_{A \text{ in } B}$

Is genome A present in my Metagenome?

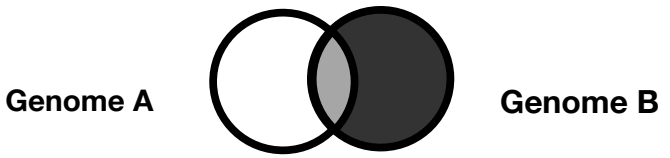


$C_{M \text{ in } A}$ ~~$C_{A \text{ in } M}$~~

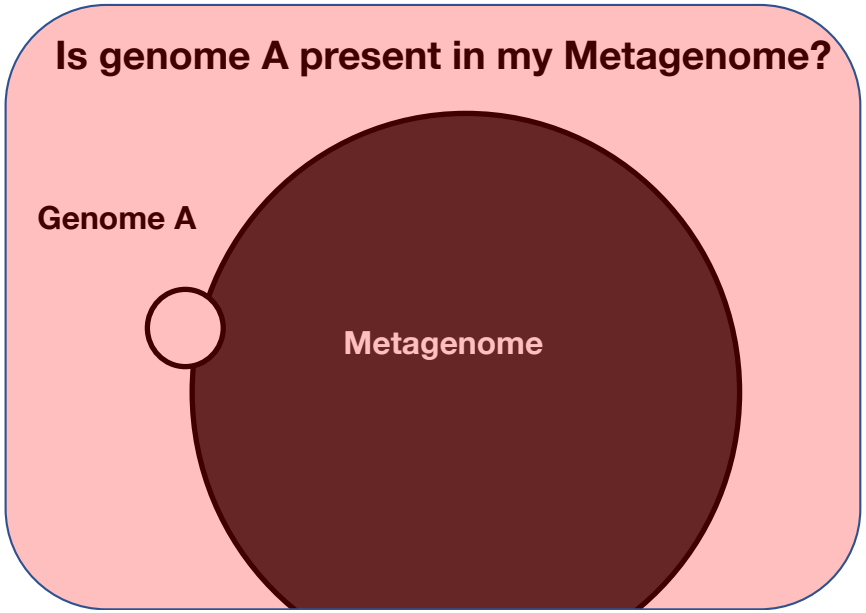
Containment => estimation of sequence similarity for a wider array of comparisons

K-mer comparisons for Sequence Similarity: Containment

How similar are my genomes?



$C_{B \text{ in } A}$ $C_{A \text{ in } B}$



$C_{M \text{ in } A}$

~~$C_{A \text{ in } M}$~~

Containment => estimation of sequence similarity for a wider array of comparisons

Prompt: How does k-mer matching to genomes compare to read mapping?

Note:

- read mapping ***places metagenome reads on a genome.***
- K-mer matching ***finds k-mers in common between a metagenome and a genome.***

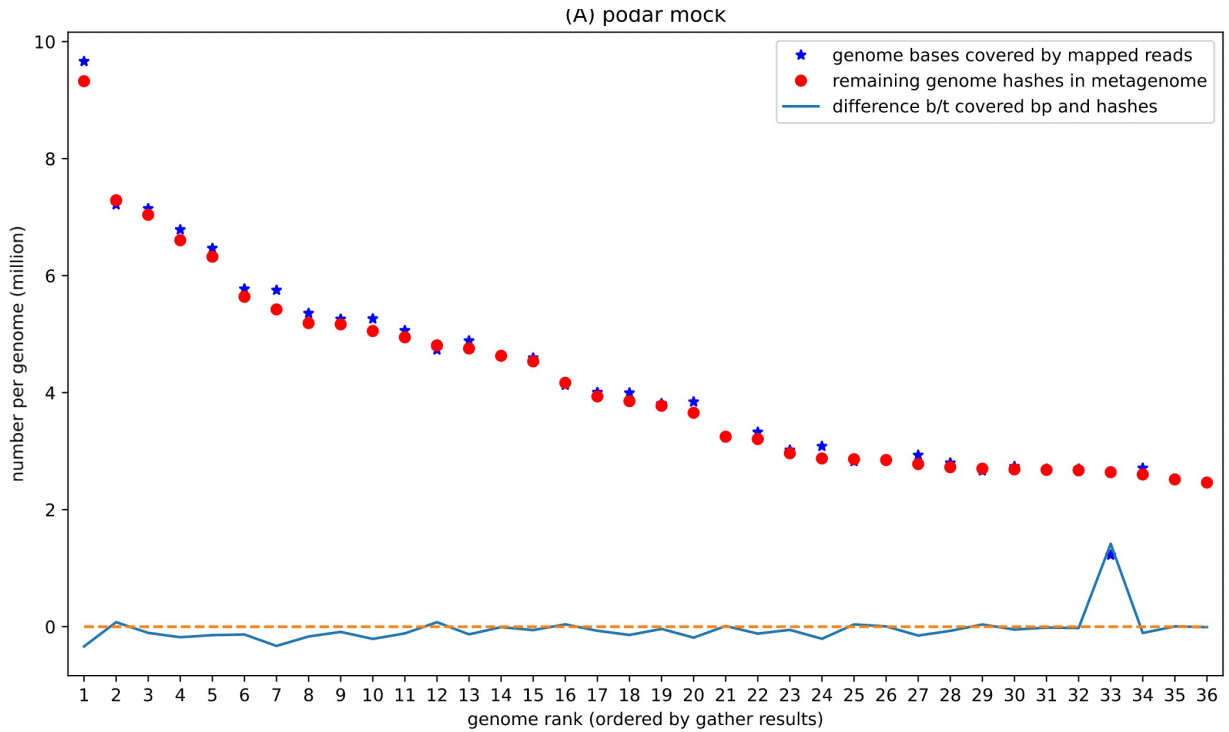
What do these numbers mean?

overlap	p_query	p_match	avg_abund	
2.0 Mbp	0.4%	31.8%	1.3	GCF_004138165.1 Candidatus Chloropl
1.9 Mbp	0.5%	66.9%	2.1	GCF_900101955.1 Desulfuromonas thio
0.6 Mbp	0.3%	23.3%	3.2	GCA_016938795.1 Chromatiaceae bacte
0.6 Mbp	0.5%	27.3%	6.6	GCA_016931495.1 Chlorobiaceae bacte
352.0 kbp	0.1%	9.3%	2.6	GCA_002440745.1 Bacteroidales bacte
306.0 kbp	0.1%	13.5%	1.5	GCA_018399635.1 Clostridia bacteriu

p_query is (abundance-weighted) fraction of metagenome 31-mers that matches to that genome; **it corresponds to fraction of reads that will map.**

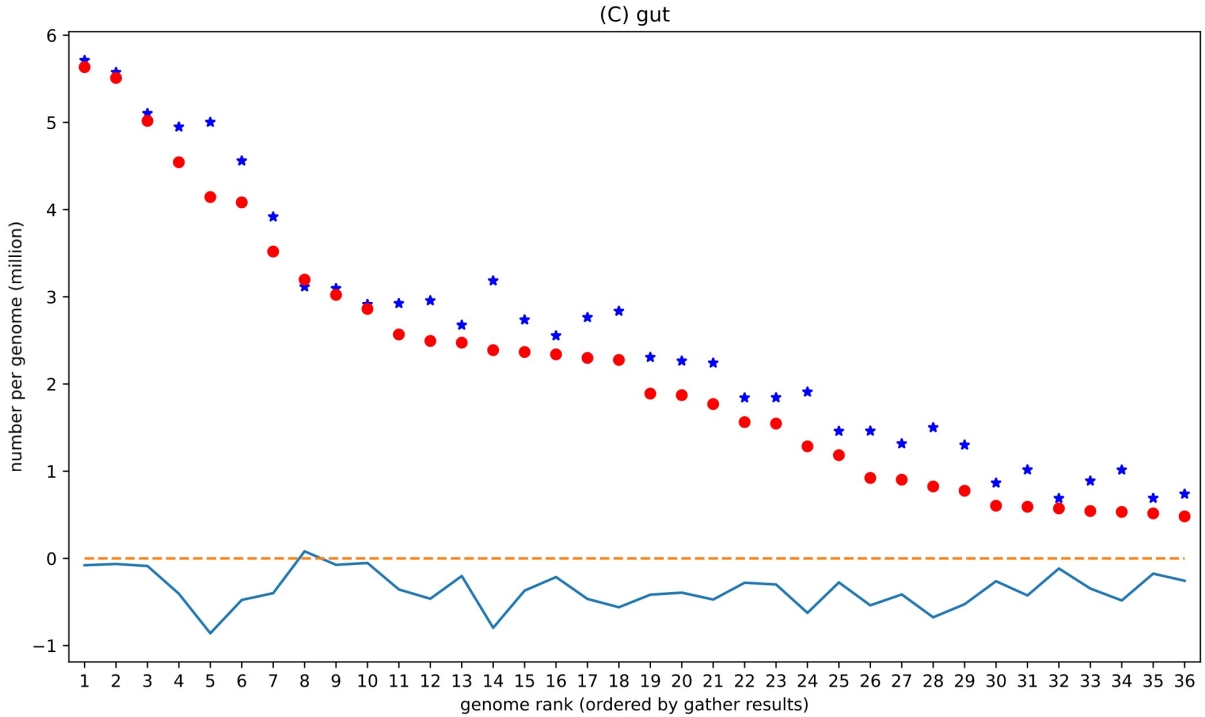
P_match is (non-weighted) fraction of genome 31-mers that are present in the metagenome; **it corresponds to fraction of covered bases in the genome.**

K-mers and read mapping correlate closely for mock communities.



Irber et al., biorxiv 10.1101/2022.01.11.475838

K-mers and read mapping correlate well for real communities, *except for* strain variation.

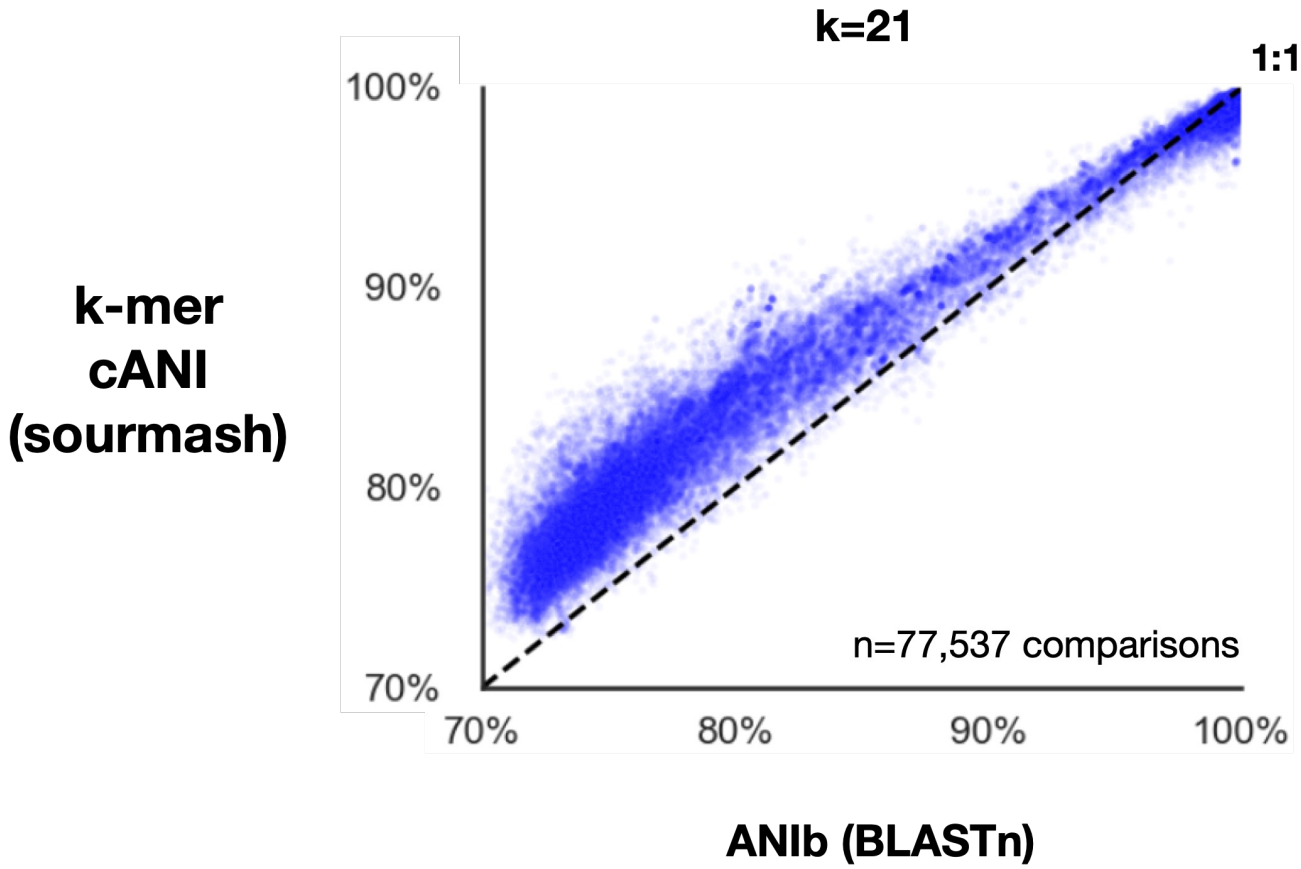


Irber et al., biorxiv 10.1101/2022.01.11.475838

Summary: k-mer matching to genomes approximates read mapping to genomes.

- K-mer overlaps between a shotgun metagenome and a genome imply that reads from the metagenome will map to that genome.
- Reference databases rarely contain exact matches for the microbial genomes present in a metagenome, but 31-mer overlaps are “good enough” to detect/choose reference genomes.

Note: k-mers can be used to estimate alignment-based stats, e.g. average nucleotide identity (ANI)



Method:

Hera, **Pierce-Ward**, and Koslicki 2022 (*bioRxiv)* 10.1101/2022.01.11.475870

Software & comparisons:

Pierce-Ward et al., in preparation.

To the hands-on tutorial!!

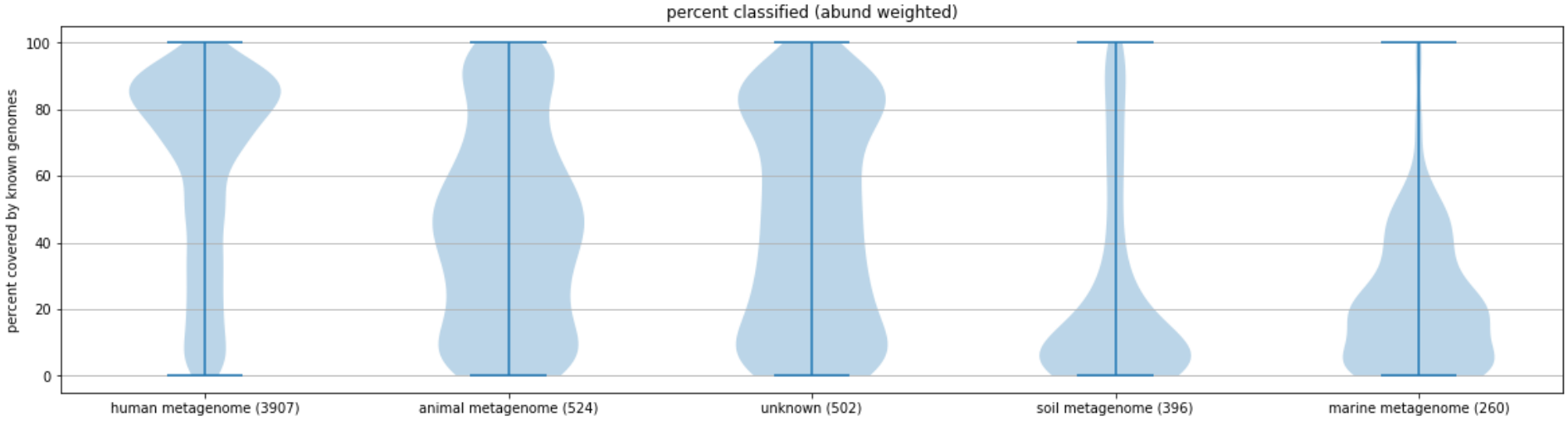
What other cool things can we do with k-mers??

K-mers let us operate on really vast scales…

Apply these things to:

- All of Genbank microbial (~1.3m genomes)
- All of SRA shotgun metagenomes (~600,000)

Preliminary results - % metagenome classified, top five metagenome categories.



https://github.com/dib-lab/2022-sra-gather

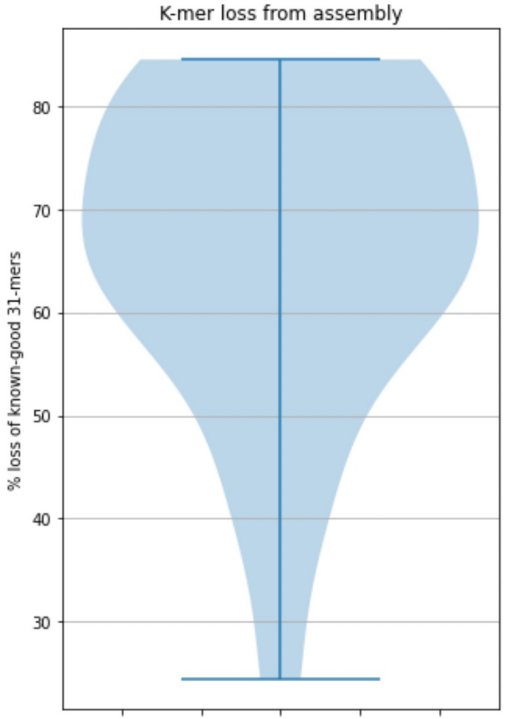
Conclusion: good reference genome sets are reliably available only for certain environments.

- Metagenomes from human microbiomes can typically map > 85% of reads to reference genomes.
- Reference genomes for other host microbiomes, or for marine/soil, are simply not available in databases.

- (Reminder: friends don't let friends study soil metagenomes 😂)

We can systematically measure loss of information from assembly, too!

~30% of “good” 31-mers are lost during metagenome assembly



Unpublished dib-lab work.

Tentative conclusion: assembly has high “false negative” rates.

- Assembly produces contigs that do not contain errors and variants present in the reads.
- This is a potentially acute problem for metagenomics, where multiple strains and variants are typically present.
- Assembly may also simply ***fail*** to produce contigs in the presence of sufficient strain variation (Awad et al., biorxiv 10.1101/155358v3)

Conclusions:

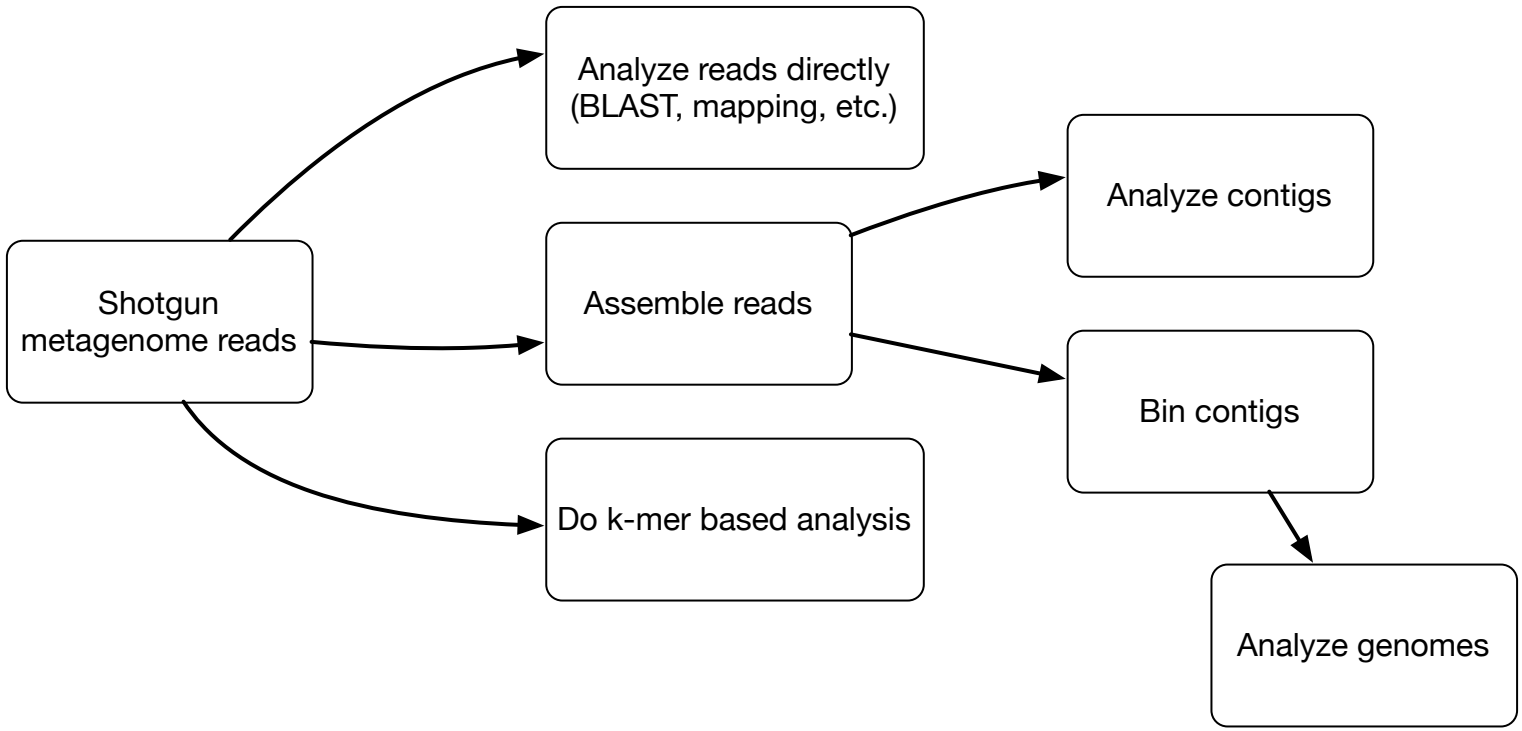
- Things that assemble are *mostly correct!!*
- But assembly *may miss a lot.* (Note, binning also has problems ☺)

Backing up a bit – what do we actually want to do here??

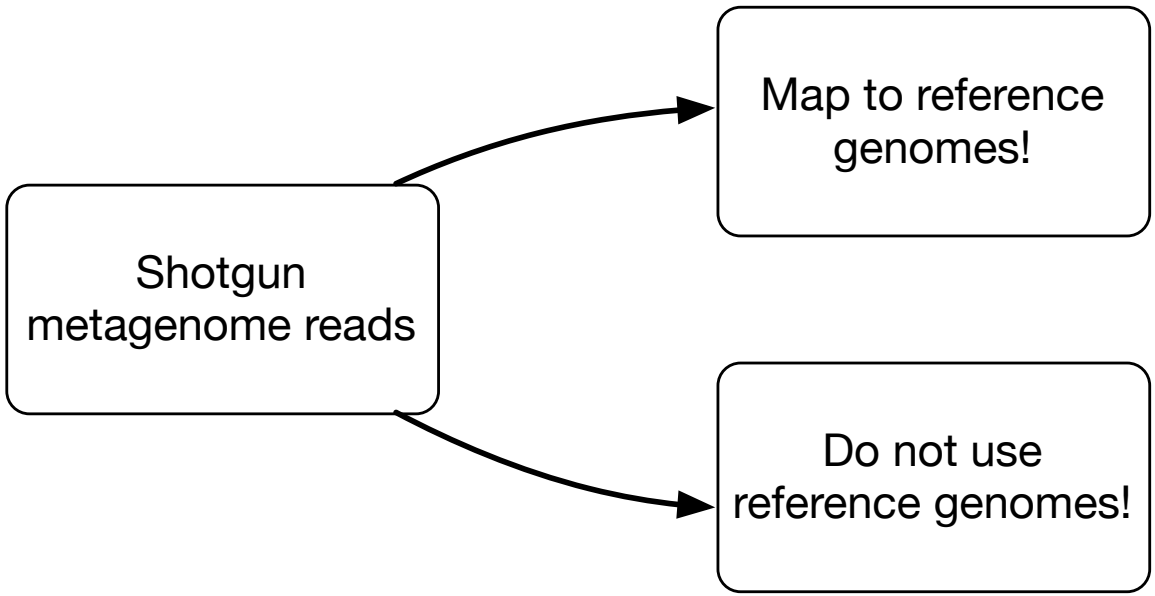
What's the goal of shotgun metagenome data analysis?

- Taxonomic/phylogenetic characterization
- **Identify potential functions.**
- **Analyze organisms w/o 16s (e.g. viruses)**

To assemble, or not to assemble?



Should you use reference genomes?



Assembly-free and reference-free are *different things*

- Assembly recovers longer contigs from shotgun sequence - both known and unknown.
- It is (usually, and most profitably IMO) a reference-free approach.
- There are *other* reference-free approaches, such as k-mer based clustering and comparison of genomes (see e.g. mash, or sourmash compare)

- *Mapping reads* to a reference genome is implicitly a reference-based approach.
- K-mers can be used flexibly – to do reference-free things (comparisons, assembly, machine learning) or reference-based things (taxonomy).

Why *not* use references?

- Reference databases are large and annoying!
  - ~1.3m bacterial + archaeal genomes in Genbank.
  - Impossible to map to all of them...
- Mapping only works if there's a species representative available
  - Need ~99% nucleotide identity to map reads reliably.
  - Outside of genus-level, genomes are too distant.!
  - Good reference genomes will often not be available for non-human metagenomes
- Accessory elements may be missing or misassigned in databases
- Reference genomes can be contaminated

Why *not* assemble/bin?

- Assembly is expensive and can be tricky
- Binning can be tricky as well
- Both assembly and binning lose information

- Official DIB Lab recommendation is:

  Do as much characterization as possible without relying on reference genomes or assembly; then salt these in as needed.

This will be all discussed in a bit more detail tomorrow.